

Taller introductorio de VHDL

IIC2343

Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación

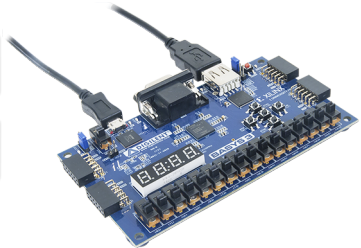
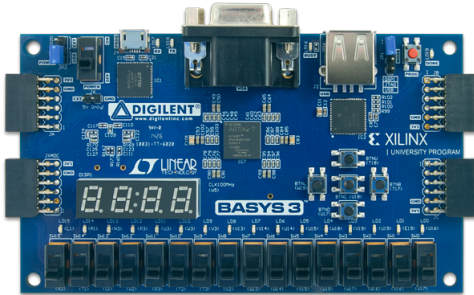
flucchini@uc.cl

August 20, 2018

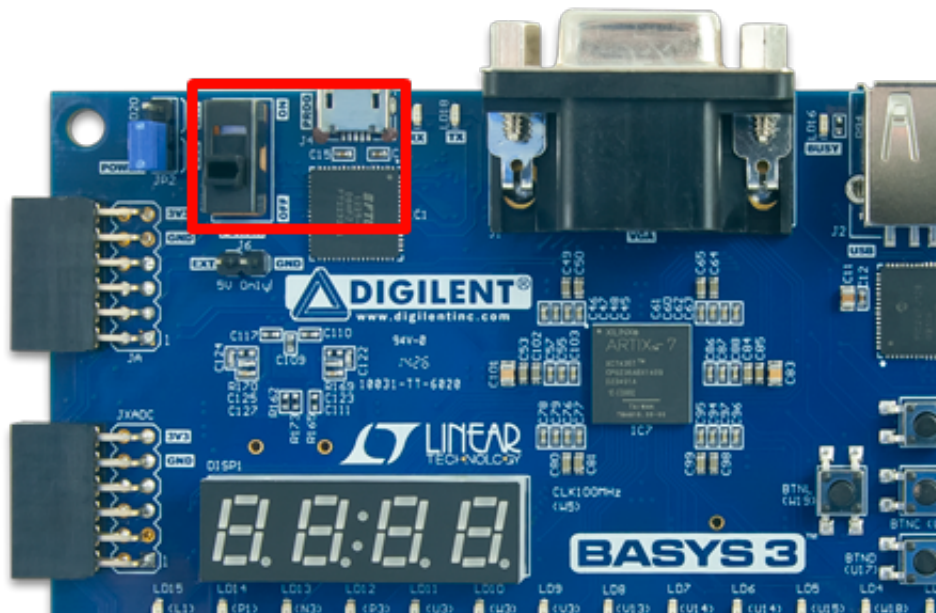
- 1 **Mostrar la placa del curso**
 - Basys 3
 - Componentes de la placa
 - Pedir una placa
- 2 **VHDL**
- 3 **Programar la Placa**
- 4 **Taller**

Basys 3

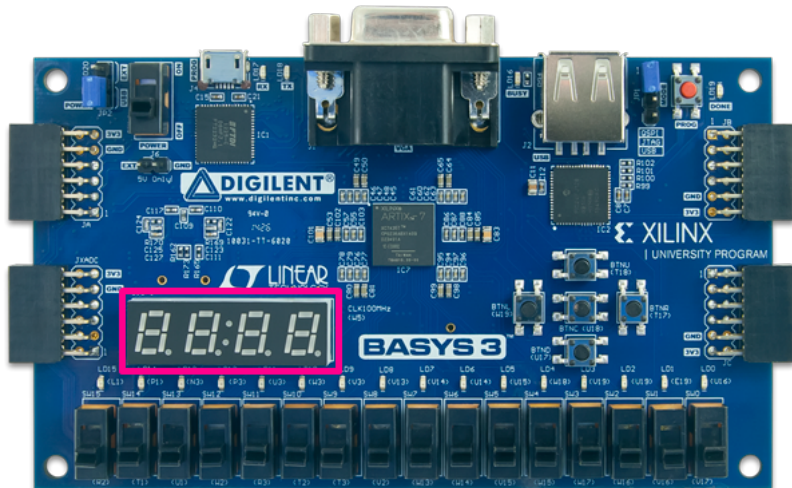
- Tratar con cuidado
- Usar cable USB micro



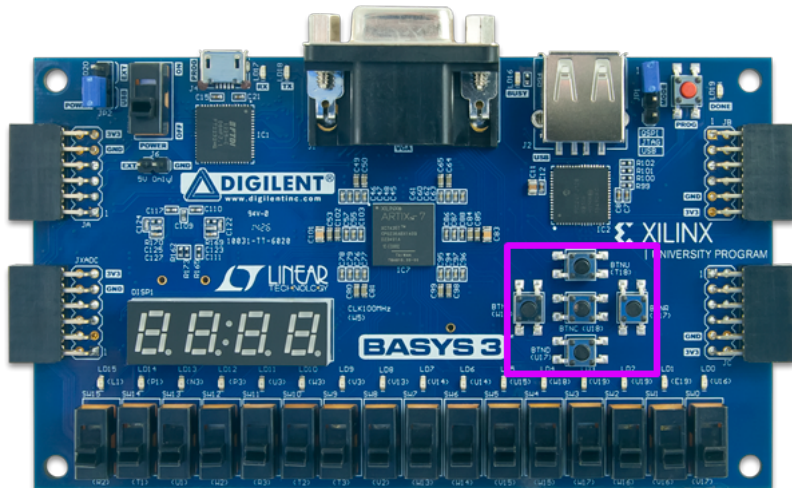
On-Off y fuente de energía



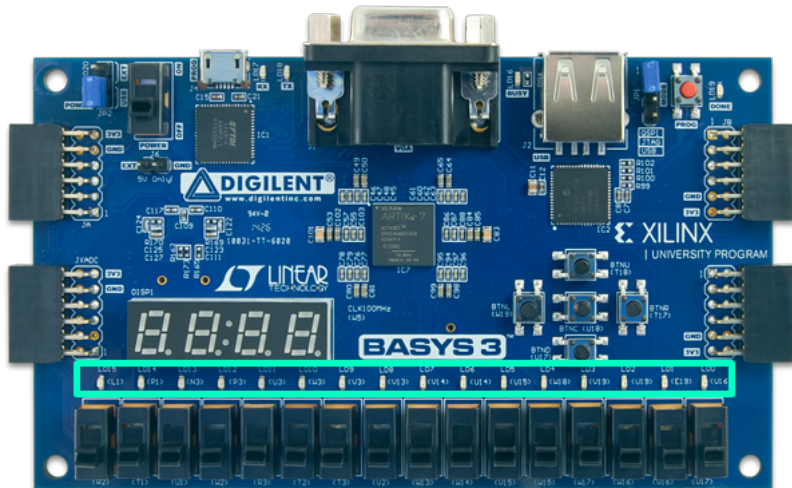
Display de 7 Segmentos



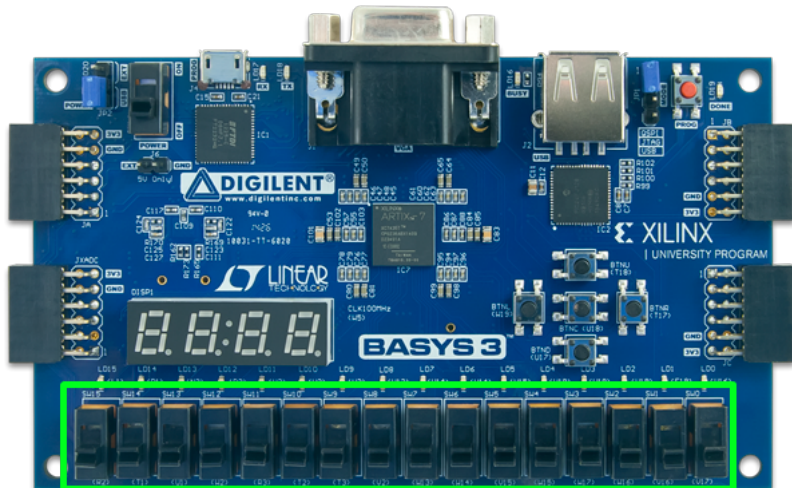
Botones



LEDs



Switches



Reglas para pedir prestada la placa:

- Las placas se pueden pedir desde las 18:30 los lunes y durante el resto de la semana.
- Sólo se puede pedir si toca versión práctica de la tarea.
- Las pueden pedir en la oficina O13 del DCC.

Reglas devolución de la placa:

- Deben devolverla a más tardar durante la ayudantía de los lunes.
- Se pueden devolver en la oficina O13 del DCC o durante la ayudantía
- El atraso de 1 día implica 1 punto menos en la nota final de la tarea.
- El atraso de más de 1 día implica nota 1.0 en la tarea.

VHDL es un programa de modelación del comportamiento de circuitos eléctricos. Tiene los siguientes elementos básicos:

- 1 Librerías
- 2 Entidades
- 3 Componentes
- 4 Señales
- 5 Operadores

Para trabajar, todos los archivos importan la siguiente librería y usan todos los elementos que provee:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

La descripción de un circuito se conoce como `entity`.

- Describen el tipo de puertos que tiene un circuito.
- Describe el comportamiento interno.
- Tiene instancias de otros componentes dentro.
- La entidad principal se conoce como `top module`.

Entidades

La descripción de un circuito se conoce como `entity`.

```
21 library IEEE;
22 use IEEE.STD_LOGIC_1164.ALL;
23
24 entity Test is
25     Port ( input_A : in STD_LOGIC;
26           input_B : in STD_LOGIC_VECTOR (15 downto 0);
27           output_A : in STD_LOGIC;
28           output_B : in STD_LOGIC_VECTOR (15 downto 0));
29 end Test;
30
31 architecture Behavioral of Test is
32
33     -- Declaración de los componentes y signals.
34
35 begin
36
37     -- Declaración de instancias y comportamientos
38
39 end Behavioral;
```

Componentes

Para usar un circuito dentro de otro, usamos instancias de components:

```
12 architecture Behavioral of FA is
13
14 component HA -- Declaracion del HA
15     Port ( a : in std_logic;
16           b : in std_logic;
17           s : out std_logic;
18           c : out std_logic);
19 end component;
```

Instancias

Ejemplo de dos instancias de un Half Adder:

```
25 begin
26
27 c <= c1 and c2;
28
29 inst_HA: HA port map( -- Primera instancia de HA
30     a      =>a, -- Conexiones de los puertos de la instancia
31     b      =>b,
32     s      =>s1,
33     c      =>c1
34 );
35
36 inst_HA2: HA port map( -- Segunda instancia de HA
37     a      =>s1,
38     b      =>c1,
39     s      =>s,
40     c      =>c
41 );
42
43 end Behavioral;
```

Las señales o `signals` se usan para representar cables y almacenamiento:

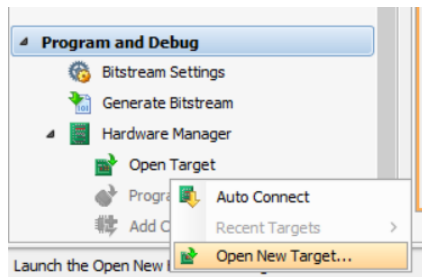
```
21 signal s1 : std_logic; -- Declaración de senales
22 signal c1 : std_logic;
23 signal c2 : std_logic;
24 signal s2 : std_logic_vector (3 downto 1);
25
26 begin
27
28 c <= c1 and c2;
```


VHDL cuenta con los operadores lógicos `and`, `or`, `xor`, `not` y de concatenación `&`

```
21 signal s1 : std_logic; -- Declaración de senales
22 signal c1 : std_logic;
23 signal c2 : std_logic;
24 signal s2 : std_logic_vector (3 downto 0);
25 signal s3 : std_logic_vector (3 downto 0);
26 signal s4 : std_logic_vector (3 downto 0);
27 signal c3 : std_logic_vector (3 downto 0);
28
29 begin
30
31 c <= c1 and c2;
32 s <= s1 or not s2(0);
33 s4 <= s2 xor s3;
34 c3 <= '1' & c2 & "01";
```

Conectar la placa

Para poder programar la placa hay que conectarla a Vivado. La primera vez, se usa Open New Target:

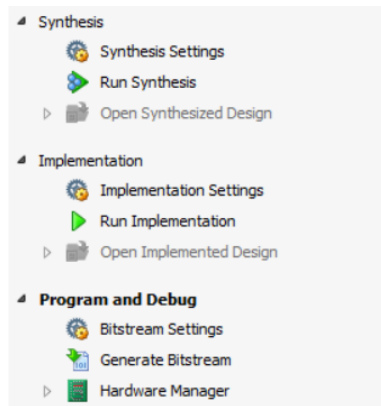


Generar el Bitstream

La placa se programa usando un archivo .bit (Bitstream), para generarlo hay 3 etapas:

- 1 Run Synthesis
- 2 Run Implementation
- 3 Generate Bitstream

En Hardware Manager, seleccionar la opción Program Device para programar la placa.



Usando el Proyecto Base y el Tutorial 2 del Syllabus, armar los siguientes componentes:

- 1 Half-Adder
- 2 Full-Adder
- 3 Sumador de 16 bits

Para probar, se puede simular el comportamiento. En el readme del Syllabus hay un link a un tutorial.